Machine Learning Challenge

The machine learning challenge is intended to be a practical and fun, cumulative project, to be completed in teams of 3-4.

We will be running a survey that asks students questions about several AI generated images. A TA will collect the survey responses from students and share the data with you in CSV file. Your task is to use the provided data to build a classifier related to this data set, e.g. to predict which AI generated image a student is referring to in their response. The exact task will be determined after the course staff has a chance to experiment with the data ourselves.

Your goal is to build a classifier that can perform well on an unseen test set that will *not* be shared with you. We will collect this test set by asking TAs and instructors to complete the same survey as students. Your model is expected to perform reasonably on this unseen data.

You are welcome to use any materials you want from this course. You can choose any model you'd like, or use an *ensemble* of multiple models, as long as your model files do not exceed 10MB in size.

One important aspect of this challenge is that the test set is not shared with you, and that your model should perform reasonably on this data. To do well, be mindful of the *bias-variance decomposition*—i.e. detecting when your model underfits/overfits.

To make this fun, there will be a prize for the group(s) who perform the best on the unseen test set! (TBD)

Machine Learning Challenge Components

The ML challenge is worth 15% of your final grade in this course. It has three components:

Data Collection Survey (1%) – Due Oct 11, 10pm

Completion of the data collection survey during the lab on October 7, 2022. This survey should take 5-10 minutes to complete. If you miss the lab, the survey can be completed by October 11, 2022, 10pm.

Model Predictions(4%) – Due Dec 5, 10pm

Submit a python3 script called `pred.py` that takes as parameter the name of a CSV file containing the test set, and produces predictions for that test set. Your script can use the following imports:

- python3
- numpy
- pandas

You will not be able to use sklearn, pytorch, tensorflow, etc in the final submitted code, so that you have a chance to build models yourselves. You may reuse any code that you wrote (or provided to you) in any of the labs. However, during the exploration phase of your project, you are welcome to use whatever tools you want before deciding what model you would like to build.

You can submit additional files that are used by `pred.py`, e.g. to store your model parameters. All of your files, taken together, should be at most 10MB in size. Your script should not require networking, and should not download any new files. Your model script should be reasonable in its use of memory resources (TBD-determine limit), and should be able to make ~60 predictions within 1 minute (TBD when we get the test set).

Your model will be graded for the following:

- **(4/4) Full credit** Model script is runnable, meets the criteria for file size, imports, and such, and produces reasonable results over the test set. We will set the "reasonable" threshold such that groups who follow good machine learning practises and chose reasonable models should be able to pass the threshold.
- **(3/4) Partial credit** Model script is runnable, meets the criteria for file size imports, and such, and produces reasonable results over the test set. The model is clearly overfit to the provided training data.
- **(2/4) Partial credit** Model script is runnable, meets the criteria for file size imports, and such, and produces reasonable results over the test set. Model choice is not reasonable and the model is clearly underfit.
- **(1/4) Runnable Script** Model script is runnable, meets the criteria for file size imports, and such, and produces results over the test set that is the output of a reasonable model (i.e. not just a constant or random output).
- **(0/4) Script not runnable** Model script is not runnable, fails the criteria for file size or imports, and such.

PDF Report (10%) – Due Dec 5, 10pm

Submit a pdf file called `csc311challenge.pdf` that describes your final model, as well as the steps that you took to develop this model. Along with the PDF file, please also submit any other `.py` or `.ipynb` files that you used while developing your model.

The `.py` or `.ipynb` files do not need to be runnable by the TA, and can rely on external imports (e.g. sklearn) and data sets that you don't submit. Only your final `pred.py` file need to be runnable.

Your report should describe:

- (2 points) Data: Describe how you explored your data, and the process through which you determined your input features. How did you end up representing your data? What else did you try? We are looking for:
  - A thorough exploration of the data, e.g. in lab 2. What are the distributions of the features? How do these features correlated with the target?
  - A clear and logical determination of how you determined your input features, with convincing logical or empirical evidence justifying your choice. Important features are not overlooked.
  - A clear description of the way(s) that you are representing the data in your models.
  - The descriptions should be consistent with the `.py` and/or `.ipynb` files that you used while developing your model.
- (2 points) Model: A description of the model(s) that you are evaluating/exploring. We are expecting a thorough exploration of at least 3+ family of models, even if you don't ultimately use those models. We are looking for:
  - How you are applying these models. You don't need to reiterate what the models are and how they work. Instead, we're looking for a description of the choices you are making to apply the model to this task.
  - A clear description of what your final model choice looks like in the submitted `pred.py` file.
  - The descriptions should be consistent with the `.py` and/or `.ipynb` files that you used while developing your model.
- (4 points) Model Choice and Hyperparameters: How you are determining what model(s) to use in your final `pred.py`, as well as an exploration of hyperparameters. We are looking for:
  - A clear description of how you are splitting your data into various sets. You may use k-fold cross validation if you'd like, but if you do, describe how you are applying that idea.
  - A clear description of the evaluation metric(s) used to evaluate your model, as well as justification for its use.
  - A description of the hyperparameters that you are tuning, hyperparameter values combinations that you have tried, and the corresponding evaluation metric(s). We are not looking for an exhaustive search of all possible hyperparameter combinations, but just enough evidence to demonstrate that your hyperparameter choices are reasonable.
  - Training curve of your final model showing the number of iterations/epochs on the x-axis, and the loss or other evaluation metric on the y-axis.
  - The descriptions should be consistent with the `.py` and/or `.ipynb` files that you used while developing your model.
- (2 points) Prediction: How well do would you expect your model to perform on the test set? We are looking for:
  - A reasoned explanation of your expected model performance, with empirical evidence supporting your explanation. You can provide a range or a point estimate (a single number) of your expected test accuracy on our data set. You are not graded on the closeness of your estimate to the final test accuracy, but you are graded
- Workload Distribution: A description of what each person in the group contributed to the project. A 1-2 sentence description of each person's role is sufficient. Each student's description must be written by that student in order for them to receive credit for the project.

Recommendations

Start early! The data exploration portion can be done as soon as data is available.

Communicate early and often. There are a lot of tasks to do as a team and decisions to be made.

Keep a journal (e.g. google doc or overleaf) and record the experiments that you perform. If you keep clear records, then writing the **Model Choice and Hyperparameter** section of your report would be much easier.

Try out various models using sklearn before writing your own implementation. Leverage the code that you have already written for the labs.

Be mindful that machine learning models can take a while to train, and that some models are more time-consuming to train than others.